

UNITED STATES PATENT APPLICATION
FOR

OBJECT MOVIE EXPORTER

INVENTORS:

Xiaochun Nie
Christopher L. Flick

Prepared by:

Blakely, Sokoloff, Taylor & Zafman
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(408) 720-8300

Attorney's Docket No. 4860P2640

"Express Mail" mailing label number: EL431887555US

Date of Deposit: August 6, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Carla Zavala

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

OBJECT MOVIE EXPORTER

BACKGROUND OF THE INVENTION

[0001] The present invention relates to methods and apparatuses for compressing, processing, transmitting, and receiving data representing multiple views of an object.

More particularly, the present invention relates to methods and apparatuses for compressing, processing, transmitting, and receiving multiple views of an object over a network of computer systems.

[0002] Digital processing systems, such as conventional computer systems, can often display various different views of an object on a display device which is coupled to the digital processing system. In many such systems, the user of the system may manipulate the object in such a way to see various views of the object. The views, in one example, may be considered to be obtained from the surface of a virtual sphere which surrounds the object. **Figure 1A** shows a virtual sphere 100 which surrounds an object 101. The different views of the object may be considered to be taken from various points on the surface of the virtual sphere 100. The virtual sphere 100 includes an equator 12 and a meridian or longitudinal line 14. Point 15B represents the North Pole of the virtual sphere and point 15A represents the South Pole of the virtual sphere. Point 16, 17, 18, and 19 on the equator 12 represent the locations 0°, 90°, 180°, and 270° respectively along the equator. If the view at point 16 along the equator 12 is considered to be a front view of the object 101, which is shown as a house, then the view from point 18 is a rear view while views from point 17 and 19 are views of the right and left sides respectively. A view from the North Pole shows the roof of the house, and a view from the South Pole

shows the bottom of the house.

[0003] Various methods exist in the prior art for manipulating such an object in order to see various views of the object. For example, U.S. Pat. No. 5,019,809 by Michael Chen describes a method for direct manipulation of an object by using a two dimensional cursor control device, such as a mouse, to simulate three-dimensional movement over the surface of a virtual sphere in order to see views of the object which is surrounded by the virtual sphere. Other methods, such as the use of sliders displayed on the screen or physical, mechanical sliders which may be manipulated by a user are also well known in the art. These various techniques allow a user to rotate or otherwise manipulate the object in order to see various different views of the object.

[0004] It is well known in the art that these views may be used to make a sequence of views which appears to be a movie. Typically, these views are displayed in a particular sequence which makes the object appear to be smoothly rotating. For example, the house 9 at the center of the virtual sphere 100 may appear to rotate on an axis defined by the north and South Pole. This "movie" is merely the playback of various selected or all of the views of the object taken along the equator 12 in sequence from point 16, through points 17, 18, and 19 back to point 16. This "movie" may be further enhanced by providing views at different latitudes.

[0005] **Figure 1B** shows an example of the various views which may be provided at each selected latitude. Table 35 includes rows 21 through 33, each of which specify at least one longitudinal 0 (along the equator) at least four views are shown; in this case, the views are from points 16, 17, 18, and 19 of **Figure 1A**. It will be appreciated that additional views may be obtained and stored to provide greater resolution along each

latitude. For example, views at every 5° or 10° along each latitude provides great resolution of the object and also makes any “movie” seem more realistic. It will also be appreciated that additional views along additional latitudes may be stored in order to provide greater resolution in the north and south directions.

[0006] Table 35 of **Figure 1B** represents a typical way in the prior art in which the various views are stored and transmitted between systems. Essentially, the views are stored in circular passes of the object at various vertical levels along the north/south axis. Typically, the physical arrangement of the data in a storage device reflects a similar arrangement of the data, which arrangement is often the manner in which the data is originally captured from the object. For example, a camera may be positioned at each of the different viewpoints in series and the data from the camera may be stored in this order such that there are essentially circular passes of the object at various vertical levels which are captured and stored on a storage device, such as hard disk or other computer readable media.

[0007] On a storage device which has random access capabilities and which provides reasonably fast rates of data retrieval, this storage arrangement provides adequate data rates such that a “movie” may be displayed from these various views. However, if this data is stored in a remote location and is accessed through a network or through a slow input/output port, then storage of this data in this arrangement does not provide adequate or satisfactory display of the object, particularly when the object is to be displayed as a “movie” which may be referred to as an “object movie”. This often happens in the case of transmission of objects through the Internet or other networks.

[0008] One major obstacle for using these types of object movies, especially with

a three-dimensional object movie, in which there are multiple views of the object, is the extremely large amount of data associated with them. To be able to transmit, store, or export the sequences of the object movies, substantial compression of the data must be accomplished. It is well known in the art that data compression is a translation of data (e.g., still images, video, audio, digital or combination) using a variety of computer compression algorithms and other techniques to reduce the amount of data required to accurately represent the content of the data.

[0009] There are at least two ways for compressing object movies, compressing every frame individually and compressing based on frame differencing. Compressing every frame individually is the same as compressing still images. For instance, JPEG compression method is one way of compressing a still image and because of that, much more space is required. Compressing based on frame differencing is accomplished by first compressing a key frame using still image compression; obtaining a delta frame, which is the difference between the current frame and the previous frame; and optionally, compressing the delta frame. Such compression continues for several subsequent frames wherein each of the subsequent frames is compared to the previous frame and a delta frame is obtained. This is typically referred to a linear compression model.

[0010] Compression of an object movie with multiple views can be done utilizing a linear sequence compression using frame differencing compression method. **Figure 1C** illustrates that a current video compression technology 102 assumes that each frame (e.g., each view) of the object movie is arranged in a linear way and compression is linearly accomplished in one direction. Each frame of the object movie represents a view of the object wherein the multiple views of the object can be taken using the method of

capturing multiple views of an object 101 described above. The frames of the object movie can be arranged in a two-dimensional array of images as shown in **Figure 1C**. As shown by the arrows pointing in direction A, compression is performed in order from frame 1 through frame 25, assuming that the object movie has 25 frames.

[0011] To compress a video sequence, for instance, with a video sequence that starts with frame 1 and ends with frame 25, the video sequence can be arranged as shown in **Figure 1C**. There are five rows in this arrangement, row 102-a, 102-b, 102-c, 102-d, and 102-e. Using frame differencing compression, a compressor usually starts from a key frame, in this example, frame 1 in row 102-a, and performs a frame differencing compression. The compressor first compresses frame 1, then, based on the difference between the current frame, frame 2, and the previous frame, frame 1, a delta frame is compressed. This event is repeated until all of the frames in row 102-a are compressed. The compressor will then continue to compress row 102-b, 102-c, 102-d and then 102-e in that order in the same manner as was done for row 102-a. (See arrows A). The number of key frames in a video sequence may be chosen by the compressor, for instance, when there is a big enough difference between two frames, the compressor will assign a key frame. Alternatively, the key frame can be defined, for instance, with a command that assign a key frame every five or ten frames in the sequence. One advantage for this compression is that the delta frame is usually smaller in size comparing to the key frame, given there is much similarity between video frames.

[0012] **Figure 1E** summarizes the current compression method 100-a for an object movie discussed above. Here, step 104 is used to capture images or views of the object (e.g., object 101 above) of the movie object at various perspectives. In step 106,

the frames representing the images are arranged and stored in a linear sequence, for example, a two-dimensional array of images. In step 108, the key frame or key frames for the video sequence is determined, for instance, by assigning a key frame to an image or a view when the image is the first frame of the sequence or, by assigning a key frame to an image or a view when there is a big enough difference between consecutive images or views. Finally in step 110, a compression method is applied to the video sequence, for example, frame differencing. The compression method is linear in compression direction in that it is compressing only in one direction.

[0013] Object movies may comprise several views hence, numerous frames. For instance, an object movie typically have hundreds of frames and even more depending on horizontal resolution (e.g., thirty pictures for each row horizontally and with eighteen rows in total, the object movie has a total of five-hundred-forty frames). The ability to enable random accessing during a user interactive experience is particularly in demand with object movies. For example, the user may wish to select views of the top the sides of the object 101 above and skip some other views. The user may also wish to designate the sequence of playback which means that the user must be allowed to access any frame in any random order. However, random accessing of frames in the current object movies compressed under the current compression method is extremely slow, lengthy, complex, tedious, and troublesome.

[0014] For the user to access a particular view, the frame to that view must be decompressed. Under the compression method 102, decompression must always start with the key frame associated with the particular frame of that view followed by decompression of as many delta frames as necessary to get to that particular view. For

example, **Figure 1D** illustrates that to access frame 7, the key frame, frame 1, must be decompressed. Then, frame 2 must be decompressed next, i.e., the delta frame between frame 2 and frame 1 must be decompressed. Then, frame 3 must be decompressed after frame 2, i.e., the delta frame between frame 3 and frame 2 must be decompressed. The decompression continues for frames 4, 5, and 6, or as many frames as necessary to get to frame 7. This sequence assumes that there is only one key frame, frame 1, between frame 1 and frame 7. Depending on the complexity and the similarities (or lack thereof) between the images taken for each object, there may be more or less key frames. As can be seen, one key frame and six delta frames needed to be decompressed before frame 7 can be decompressed and accessed. Similarly, if frame 25 needed to be accessed, one key frame and twenty-four delta frames needed to be decompressed before frame 25 can be decompressed and accessed.

[0015] Because linear compression is typically a one-direction compression, decompression is slow and not optimized. As illustrated, numerous steps of decompression are thus necessary thereby slowing down the random access interactivity. Appointing more key frame in a video sequence can minimize steps of decompression. However more key frames means that the compressed file will be bigger resulting in slow compression rate, transmission rate and exporting rate. Furthermore, the data will be more costly to generate.

[0016] It is thus desirable to have compression methods that enable quick and simple decompression step while keeping the cost of the method low.

[0017] Demands for an efficient compression and decompression rate have grown even more. The modern trend is that users typically request for views of an object over

the Internet. **Figure 2A** shows several computer systems which are coupled together through the Internet 103. It will be appreciated herein that the term "Internet" refers to a network of networks which uses certain protocols (e.g. the TCP/IP protocol and possibly other protocols such as HTTP (hypertext transfer protocol) for HTML (hypertext markup language) documents). The physical connections of the Internet and the protocols and communication procedures of Internet are well known to those in the art. Access to the Internet 103 is typically provided by Internet service providers (ISP's) such as ISP's 105 and 107. Users on client systems, such as client computer systems 121, 125, 135, and 137 obtain access to the Internet through the Internet service providers. Access to the Internet allows users of the client computer systems to exchange information, to receive and send e-mails, and to view and manipulate these objects as they are received. For example, web server system 109 may contain data representing the object 101 shown in **Figure 1A** and provide this data to a client computer system such as client system 121 upon request by the client system 121. Often these web servers are provided by ISPs, such as ISP 105, although a computer system may be set up and connected to the Internet without that system also being an ISP as is well known in the art.

[0018] The web server system 109 is typically at least one computer system which operates as a server computer system and is configured to operate with the protocols of the World Wide Web (WWW) and is coupled to the Internet. Optionally, the web server 109 may be part of an ISP which provides access to the Internet for client systems. The web server 109 is shown coupled to other computers in the Internet 103. Client computer systems 121, 125, 135, and 137 may each, with the appropriate web browsing software, view HTML pages provided by the web server 109. These web pages

may provide movies, such as QuickTime movies, which may be viewed by users of the particular client computer system.

[0019] The ISP 105 provides Internet connectivity for the client computer system 121 through the modem interface 123 which may be considered part of the client computer system 121. The client computer system may be a conventional computer system such as a Macintosh computer, a “network” computer, a Web TV system, or other types of digital processing systems, such as a cellular telephone having digital processing systems or capabilities. Similarly the ISP 107 provides Internet connectivity for client systems 125, 135, and 137, although as shown in **Figure 2A**, the connections are not the same for these three computer systems. Client system 125 is coupled through a modem interface 127 while client computer systems 135 and 137 are part of a Local Area Network (LAN). While **Figure 2A** shows the interfaces 123 and 127 as a modem, it will be appreciated that each of these interfaces may be an analog modem, an ISDN modem, a cable modem, a satellite transmission interface (e.g. “Direct PC”), or other interfaces for coupling a computer system or a digital processing system to other digital processing systems. Client computer systems 135 and 137 are coupled to a LAN bus 133 through network interfaces 139 and 141 which may be an Ethernet network interface or other network interfaces. The LAN bus is also coupled to a gateway computer system 131 which may provide firewall and other Internet related services for the local area network. This gateway computer system 131 is coupled to the ISP 107 to provide Internet connectivity to the client computer systems 135 and 137. The gateway computer system 131 may be conventional server computer system. Also, the web server system 109 may be a conventional server computer system.

[0020] Even with modern, high-speed analog modems, data transmission rates through the Internet are often painfully slow. Thus, a user of a client system may request various views representing an object to allow the user to inspect the object or to manipulate the order of viewing the object. This request will be processed by a server system or some other digital processing system and the data will be transmitted to the requesting client system. This data will be transmitted to the client system and decompressed, for example, in the order shown in **Figures 1C-1D** which is typically also the same order used to play back a movie of the object. For example, a series of views along the equator beginning at 0° and progressing consecutively at 5° increments back to 0° may be transmitted from the server system to a client system. The user may request views in any particular order along the virtual sphere surrounding the object. Decompressing each of these views, which are often high-resolution digital data, so that these data can be transmitted, can take a considerable amount of time.

[0021] The random access of the data for the "object movie" requires sequential decompression of all the prior frames as illustrated in **Figure 1D**. The user must patiently wait for the completion of the decompression of all of the other prior frames for each of the selected frames. The interactive experience is thus painfully slow. Therefore, it is desirable to provide methods and apparatuses for improved compression and transmission of data representing views of an object.

SUMMARY OF THE INVENTION

[0022] The present invention discloses methods and apparatuses for compressing and transmitting data representing different views of an object. These methods and

[0028] Computer systems that practice the methods of the invention are also described. Further, computer readable media having software which allows the computer systems to perform the methods of the present invention are described.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] **Figure 1A** illustrates a virtual sphere which surrounds an object.

[0030] **Figure 1B** illustrates a table indicating the storage of data for an object according to the prior art.

[0031] **Figure 1C** illustrates a compression method for an object movie of the current art.

[0032] **Figure 1D** illustrates a decompression sequence of a frame of a particular view of an object movie of the current art.

[0033] **Figure 1E** illustrates a flow chart of a compression method for an object movie of the current art.

[0034] **Figure 2A** illustrates an example of several client computer systems coupled to the Internet.

[0035] **Figure 2B** illustrates an example of a computer system which may be used with the present invention.

[0036] **Figure 3A** illustrates an exemplary method of capturing images of an object for an object movie of the present invention.

[0037] **Figure 3B** illustrates an object movie of a statute using an exemplary method of capturing images of an object of the present invention.

[0038] **Figure 4A** illustrates an exemplary embodiment of method to compress a

5 x 5 object movie of the present invention.

[0039] **Figure 4B** illustrates an exemplary embodiment of a reference track and a data source track of the present invention.

[0040] **Figure 4C** illustrates a decompressing exemplary embodiment according to the present invention.

[0041] **Figure 5A** illustrates an exemplary preferred layout for a 14 x 9 object movie.

[0042] **Figure 5B** illustrates an exemplary embodiment of the present invention of a method of dividing a preferred layout into a plurality of blocks and methods to compress each of the blocks.

[0043] **Figures 5C and 5C-1** illustrate exemplary reference tracks and data source track for the 14 x 9 object movie shown in **Figure 5A**.

[0044] **Figure 6** illustrates an exemplary flowchart indicating an example of a method of compression according to the present invention.

[0045] **Figure 7** illustrates an exemplary flowchart indicating an example of a method of creating an object movie and compressing the movie object according to some embodiments of the present invention.

[0046] **Figure 8A** illustrates an exemplary flowchart indicating an example of a method which may be practiced in a system having, in one example, a client and a server according to the present invention.

[0047] **Figure 8B** show a method which may be practiced by a receiving digital processing system, such as a client computer system.

[0048] **Figure 9** illustrates an example of a computer readable storage medium

for a client computer system which may be used with one aspect of the present invention.

[0049] **Figure 10** illustrates an example of a computer readable storage medium for a server computer system which may be used with one aspect of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0050] The subject invention will be described with reference to numerous details set forth below, and the accompanying drawings will illustrate the invention. The following description and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to provide a thorough understanding of the present invention. However, in certain instances, well known or conventional details are not described in order to not unnecessarily obscure the present invention in detail. In the drawings, the same element is labeled with the same reference numeral.

[0051] **Figure 2B** shows one example of a conventional computer system which may be used as a server computer system or as a client computer system or as a web server computer system. It will be appreciated that such a computer system may be used to perform many of the functions of an Internet service provider, such as ISP 105. The computer system 151 interfaces to external systems through a modem or network interface 169. It will be appreciated that the modem or network interface 169 may be considered part of the computer system 151. This interface 169 may be an analog modem, an ISDN modem, a cable modem, a token ring interface, a satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a digital processing system to other digital processing systems.

[0052] The computer system 151 includes a processor 153 which may be a conventional microprocessor, such as a Motorola PowerPC microprocessor or an Intel Pentium microprocessor. Memory 155 is coupled to the processor 153 by the bus 157. Memory 155 may be dynamic random access memory (DRAM) and may also include static RAM (SRAM). The bus 157 couples the processor 153 to the memory 155 and also to mass memory 163 and to display controller 159 and to the I/O (input/output) controller 165.

[0053] Display controller 159 controls in the conventional manner a display on the display device 161 which may be a CRT or a liquid crystal display device. The input/output devices 169 may include a keyboard, disk drives, printers, a scanner, a digital camera, and other input and output devices, including a mouse or other pointing device. The display controller 159 and the I/O controller 165 may be implemented with convention well known technology.

[0054] The mass memory 163 is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory 155 during execution of software in the computer system 151. It will be appreciated that the computer system 151 is one example of many possible computer systems which have different architectures. For example, Macintosh or Wintel systems often have multiple busses, one of which may be considered to be a peripheral bus. Network computers may also be considered to be a computer system which may be used with the present invention.

[0055] Network computers may not include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory 155 for

execution by the processor 153. A Web TV system, which is known in the art, may be considered to be a computer system according to the present invention, but it may not include certain features shown in **Figure 2B**, such as certain input or output devices. A cell phone having a suitable display and a processor and memory may also be considered to be a digital processing system or a computer system which may be used with the present invention. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

[0056] It will also be appreciated that the computer system 151 is typically controlled by an operating system software which includes a file management system, such as a disk operating system, which is part of the operating system software.

[0057] **Figure 3A** illustrates an example of how an object movie can be created. An object can be captured or created using the conventional method discussed above. Typically, the object movie is thought of as a three-dimensional object movie because one can see all of the views around the object giving it a three-dimensional appearance. More particularly, one way to think about object movies is as a set of individual views taken from the surface of an invisible sphere 360 which surrounds the object. It is useful to imagine a consistent coordinate system of points evenly spaced apart around the surface of the sphere as illustrated in **Figures 1 and 3**.

[0058] In one embodiment of the present invention, any type of camera 352 can be used to capture the views for the object, here, the object is a cup 350. In one embodiment, at each junction of coordinates on the surface of the sphere, an image of the object is captured. The vertical position, or tilt, looking directly down on the cup 350 would have a 90 degrees vertical angle, while the vertical position looking directly up at

the object from below would be a -90 degrees vertical angle. The mid-point vertically (i.e., the equator) would be 0 degrees. Images of the cup 350 along the vertical position can be captured by tilting the camera 352 to different angles starting from a particular point along the vertical position. Similar to each of the vertical angle positions (tilts) from 90 degrees to -90 degrees, a similar sequence of horizontal positions (pans) which would range from 0 degrees to 360 degrees around the sphere can be used to captures images of the cup 350.

[0059] The number of degrees between each of the vertical and horizontal angles is up to particular applications or preferences of the developer. In one embodiment, it is preferred that both the horizontal and the vertical angles be spaced similarly. Thus, if the images are captured every 10 degrees vertically, the images should also be captured every 10 degrees horizontally. Similar angles spacing helps ensure that the resulting object movie has a consistent responsiveness to cursor movement and changes from view to view in any direction in a uniform manner. **Figure 3B** illustrates an example of an object movie 370 showing various images of a statute having captured using the method described above.

[0060] In one embodiment, the views captured around an object remain as individual still images. After capture, each individual image is incorporated as a single frame of the object movie. As well known in the art, a frame is an individual image of the object movie. The frames are then arranged into a preferred layout.

[0061] **Figures 4A-4C** illustrate a method of compression according to the present invention that will not only optimize the compression paths but also minimize the size of a final compressed file size. **Figure 4A** illustrates a compression method 400 for

between one frame to the next is not dramatic. Examples of such characteristics and features include backgrounds or colors of the object.

[0064] For a larger size object movie, the preferred layout should be divided into a plurality of blocks (see below). In this example, the compression paths can be established using the following rules. (1) Always start compressing with the key frame (e.g., f-13). (2) Always sweep away from the key frame. (3) Sweep horizontally or vertically from the key frame until reaching the boundaries 411-414 of the object movie to cover all of the frames possible in the horizontal path 415 and the vertical path 416. (4) From the horizontal path 415 and the vertical path 416 of each frame, sweep diagonally (e.g., diagonal paths 417-428) until reaching the boundaries to cover all of the frames not reached by the horizontal path 415 and vertical path 416. And, (5) always sweep such that the compression paths reach every one of the frame in the object movie. The rules above result in multi-directional compression paths. Because of the multi-directional sweeping in the compression method, the random access to any of the frames in the movie object is easy, quick and convenient.

[0065] The discussion below demonstrates some compression paths for the object movie discussed in **Figure 4A**. Compression paths for the row 401 are: compress frame f-13, the delta frame between f-13 and f-7, and the delta frame between f-7 and f-1 to get frame f-1; compress frame f-13, the delta frame between f-13 and f-8, and the delta frame between f-8 and f-1 to get frame f-2; compress frame f-13, the delta frame between f-13 and f-8, and the delta frame between f-8 and f-3 to get frame f-3; compress frame f-13, the delta frame between f-13 and f-8, and the delta frame between f-8 and f-4 to get frame

f-4; lastly, compress frame f-13, the delta frame between f-13 and f-9, and the delta frame between f-9 and f-5 to get frame f-5.

[0066] Compression paths for the row 402 are: compress frame f-13, the delta frame between f-13 and f-12, and the delta frame between f-12 and f-6 to get frame f-6; compress frame f-13 and the delta frame between f-13 and f-7 to get frame f-7; compress frame f-13 and the delta frame between f-13 and f-8 to get frame f-8; compress frame f-13 and the delta frame between f-13 and f-9 to get frame f-9; lastly, compress frame f-13, the delta frame between f-13 and f-14, and the delta frame between f-14 and f-10 to get frame f-10.

[0067] Compression paths for the row 403 are: compress frame f-13, the delta frame between f-13 and f-12, and the delta frame between f-12 and f-11 to get frame f-11; compress frame f-13 and the delta frame between f-13 and f-12 to get frame f-12; compress frame f-13 to get frame f-13; compress frame f-13 and the delta frame between f-13 and f-14 to get frame f-14; lastly, compress frame f-13, the delta frame between f-13 and f-14, and the delta frame between f-14 and f-15 to get frame f-15.

[0068] Compression paths for the row 404 are: compress frame f-13, the delta frame between f-13 and f-12, and the delta frame between f-12 and f-16 to get frame f-16; compress frame f-13 and the delta frame between f-13 and f-17 to get frame f-17; compress frame f-13 and the delta frame between f-13 and f-18 to get frame f-18; compress frame f-13 and the delta frame between f-13 and f-19 to get frame f-19; lastly, compress frame f-13, the delta frame between f-13 and f-14, and the delta frame between f-14 and f-20 to get frame f-20.

Here, frame f-8 would not be compressed multiple times according to the scheme above but instead, frame f-8 would be compressed only one. Thus, each time frame f-8 is needed in the subsequent compression paths, the compressor would have memorized that frame f-8 has already been compressed and skip the compression on frame f-8.

[0071] Further, a data source track 470 as shown in **Figure 4B** is generated. The data source track 470 stores the actual compressed data for each of the frames in which, each of the frames is only compressed once. The compressed key frame and the compressed non-key frames can be stored here and be referred to multiple times. A video track 450 is also generated. The video track 450 is a reference track that stores all of the references to the sequences of compression (or decompression) for each of the frames in the object movie.

[0072] A reference table (not shown) or reference links 451-465 (as shown in **Figure 4B**) are generated to refer each of the frames and its compression (or decompression) sequence in the video track 450 to the data source track 470. In one example, the links are incorporated into an edit list of a QuickTime™ Object Movie. (QuickTime is a registered trademark of Apple Incorporation). The edit list is part of the QuickTime object movie file format. This list is a mechanism for the video track 450 to reference the compressed data from the data source track. It is a table that records the linking between the video track 450 and the data track 470. The video track 450 thus, only indicates the compression and the decompression paths for each of the frames in the object movie. The video track 450 only contains the order or the sequence to compress and decompress each of the frame but not the actual compressed data themselves. The data source track 470 comprises all of the actual compressed data. The links 451-465 can

be used to link the compression or decompression sequence to the compressed data in the data source track 470 as many times as necessary. To get the compressed data for any one frame, the links 451-465, for example, would link the compression or decompression sequence in the video track 450 to the actual compressed data. This way, the compressed data can be accessed easily and randomly on a per need basis.

[0073] For instance, the compressor compresses frame f-13, the delta frame between f-13 and f-8, and the delta frame between f-8 and f-2 to compress f-2. During this compression process, frame f-13, f-8 and f-2 have been compressed. When the compressor continues to compress subsequent frames, the compressor will skip the compression that requires the compression of frames f-13, f-8, and f-2 again. For instance, to compress f-3, the compressor will skip f-13, and the delta frame between f-13 and f-8, and only compress the delta frame between f-8 and f-3. This process continues until all of the frames are compressed. Then, to randomly access any one frame, (e.g., f-3) the decompressor would ascertain the sequence of decompression of that frame by examining the video track 450 (e.g., f-13, f-8, and f-3). Using the links (e.g., 455, 458, and 460), the video track 450 will lead the decompressor to the location in the data source track 470 that contain the compressed frames and delta frames needed to decompress frame f-3.

[0074] Another feature of the present invention is that the compression path is a multi-directional path. Furthermore, because of the implementation of the video track 450 that can reference the data source track 470 multiple times, the compression and decompression processes does not involve redundant steps. The final compressed file thus remains small.

[0075] Another feature of the present invention is that the video track 450 and the data source track 470 can be stored in different places, for example, in two different files. This allows for the flexibility in the data storage and maintenance. The links or the reference table can be in the same file with the video track 450 to allow quick reference and direction for the random access of the frames in the object movie track 450.

[0076] In a preferred embodiment, the preferred layout, the arrangement of the blocks and the selection of the key frame are such that the decompression sequence for any one frame is less than three steps. As shown in **Figure 4A-4C**, the decompression sequence 480 for the last frame, frame f-25 of the 5 x 5 movie object is only a three-step decompression process (e.g., decompress frames f-13, f-19, and then f-25). **Figure 4C** also shows another advantage that the present invention has. Contrary to the seven-step decompression process of the conventional art as illustrated in **Figure 1C**, the decompression sequence 470 of the present invention for frame f-7 is only a two-step process. For instance, to get frame f-7, all that is needed to be done is to decompress frame f-13 and then decompress f-7, which is obtained by decompressing the delta frame between f-13 and f-7. The decompression for frame f-7 in this example assumes that there is only one key frame, f-13 between frames f-13 and f-7. Therefore, in this preferred embodiment, there are only a maximum number of two frames between any one non-key frame and the key frame.

[0077] It will be appreciated that in another implementation of the embodiment discussed above, the maximum number of frames between any one non-key frame and key frame is three. This is for optimization reason, theoretically, there is no limit for the maximum number. Therefore, the number of frames between any one non-key frame and

key frame can be set at a predetermined number of frames depending on the particular characteristic of the object movie.

[0078] **Figure 5A** illustrates a larger size object movie than the one described in **Figure 4A**. The frames are arranged in a preferred layout 500, which is also a two-dimensional array. And, each of the frames is assigned a reference number. As shown, the preferred layout 500 includes nine rows, 515-523 and fourteen columns, 501-514. Row 515 includes frames f-1 to f-14, row 516 includes frames f-15 to f-28, row 517 includes f-29 to f-42, and so forth; and, column 501 includes f-1, f-15, f-29, f-43, f-57, f-71, f-85, f-99, and f-113, column 502 includes f-2, f-16, f-30, f-44, f-58, f-72, f-86, f-100, and f-114, and so forth.

[0079] With a conventional method, the object movie of **Figure 5A** would have to be compressed linearly as discussed above. The compressor would usually start from a key frame and then based on the difference between the current frame and the next frame, a delta frame is compressed. The compressor would continue to scan for the difference between one current frame to the previous and compress the delta frame between those two frames until the compressor encounters another key frame. In that event, the process just described starts for the next set of frames that share similarities with that new key frame. Often, the compressor would always start from the key frame and sweep in one direction until it covers all of the frames in the object movie.

[0080] Using the conventional method, decompression for random access during a user interactive activity would be slow and complicated because for every frame that the user wishes to access, all of the delta frames before that frame leading up to the key frame and including the key frame must be decompressed. If the user wishes to access a

non-key frame that is further away from the key frame, the decompression would take a very long time.

[0081] In an exemplary embodiment of compression method 500-1 of the present invention, the preferred layout 500 is divided into a plurality of blocks, block B-1, block B-2, block B-3, block B-4, block B-5, and block B-6. (**Figure 5B**). Preferably, the preferred layout 500 is divided such that the frames in each of the blocks represent particular regions of the object and the blocks are arranged such that they are proximately associated with each other. Thus, similarities between rows and columns are both taken into account. Furthermore, it is also preferred that the frames in each of the blocks share spatial similarities with adjacent frames so as to minimize the number of key frames needed for each block.

[0082] Each block can be of a predetermined size for example, 3 x 3, 3 x 4, 4 x 4, 5 x 4, 5 x 5, 4 x 6, 5 x 6, 6 x 6, 6 x 7, 7 x 7, etc. In this particular embodiment, the block B-1 is a 4 x 4 block; B-2 and B-3, 5 x 4; B-4, a 4 x 5; and B-5 and B-6, 5 x 5.

[0083] In another embodiment, the preferred layout is divided such that the smaller size block is closer to the top and left corner of the layout and that larger size block is closer to the bottom and right corner of the layout. As shown in **Figure 5B**, block B-1 having a block size of 4 x 4 is at the top left corner C-1 whereas block B-6 having a block size of 5 x 5 is at the bottom right corner C-4 of the preferred layout 500.

[0084] By dividing the preferred layout into a plurality of blocks, the compression paths can be shortened for each frame. **Figure 5B** further illustrates a compression method of the present invention. For each of the blocks, B-1, B-2, B-3, B-4, B-5, and B-6, at least one key frame is selected. The more spatial similarities between the adjacent

frames in each of the block, the less likely that more than one key frame is necessary. In one embodiment, frame f-16 is chosen to be the key frame for block B-1. A key frame can be a complete frame that represents the view of the block, here, the block B-1. In frame differencing compression, other non-key frames of the block B-1 will be compared to the key frame. Each time a frame with differences from the frame before it comes along, the differences are calculated from the frame before it, which is calculated from the frame before it, and so on, back to the key frame.

[0085] In one embodiment, once the key frame for each of the blocks of the preferred layout 500 is determined, the compressor of the compression method 500-1 compresses all of the frames in the blocks. Each of the blocks is compressed separately using any conventional compression technique. In this way, instead of compressing the frames of a big layout, each of the blocks can be viewed as a small subset of the object movie. The compression of each of the blocks is thus, performed separately.

[0086] In another embodiment, during compression, if a frame is a non-key frame, its previous frame must be compressed first. This is equivalent to finding a unique path that starts from a key frame and ends at a certain frames in the block. For instance, in compressing the block B-1, the compression paths may be initiated from the key frame f-16. Similar to the compression method 400, one compression rule of the present embodiment is that the compressor will perform a walk-away from the key frame in each block to perform frame differencing (or frame predicting) compression method until the boundaries of each of the blocks are reached. The walk-away from the key frame rule may require that the compressor will sweep horizontally or vertically from the key frame until it reaches the boundaries. Furthermore, from each frame of the block, the

compressor may sweep diagonally until reaching the boundaries. In this sense, the compression method of the present invention is non-linear in that the compressor sweeps in more than one direction. Because of the multi-directional sweeping in the compression method, the random access to any of the frames in the movie object that has a few hundreds frames can be easy, quick and convenient. This is yet another advantage over the conventional compression. It also dramatically reduces the number of key frame, which will reduce the file size.

[0087] As can be seen from **Figure 5B**, a compression path for frame f-1 is compressing frame f-16 and then the delta frame between frame f-16 and f-1. A compression path for frame f-2 is compressing frame f-16 and then the delta frame between frame f-16 and f-2. A compression path for frame f-3 is compressing frame f-16 and then the delta frame between frame f-16 and f-3. A compression path for frame f-4 is compressing frame f-16, then, the delta frame between frame f-16 and f-17, and then, the delta frame between frame f-17 and f-4. A compression path for frame f-15 is compressing frame f-16 and then the delta frame between frame f-16 and f-15. A compression path for frame f-17 is compressing frame f-16 and then the delta frame between frame f-16 and f-17. A compression path for frame f-18 is compressing frame f-16, then, the delta frame between frame f-16 and f-17, and then, the delta frame between frame f-17 and f-18. The compression paths for each of the frames in the block B-1 is illustrated by the dark arrow lines. As can be seen, for any one frame, the longest compression path is two frames between any non-key frame and the key frame.

[0088] In another embodiment, to further optimize the compression path, the key frame for each of the block is selected to be the center frame, or the frame that is closest

to the center of the block. If there is no center frame, for example as can be seen in blocks B-1 and B-4, the frame closest to the center of the block is chosen. Furthermore, it is preferred that the key frame is chosen such that it is closer to the top and left corner of the block. For instance, in block B-1, the key frame f-16 is selected to be close to the top left corner C-5 of the block B-1.

[0089] One advantageous feature of the present invention is that it enables high compression rate, quick decompression time, and short compression and decompression sequence for each of the frames in the block. The compression paths in the object movie according to the embodiments of the present invention are optimized. Since the key frame is a center frame or almost a center frame, this embodiment also takes advantages of the spatial similarities between the key frame and the frames surrounding the key frame. And, combining that with the multidirectional compression paths, the compression path for any frame in each of the blocks is short and simple.

[0090] The compression paths for the blocks B-2, B-3, B-4, B-5, and B-6 can be similar to the compression paths for the block B-1 discussed above and as illustrated in **Figure 5B**.

[0091] Similar to **Figures 4A-4B**, **Figures 5A-5B** illustrate the compression paths for a larger object movie with the preferred layout 500. In one embodiment, even though there are six distinguished blocks, (as shown in **Figure 5B**) the frames are labeled as if there is only one large block (as shown in **Figure 5A**). The frames are labeled starting from row 515 and in one direction until the last frame in row 523 is labeled. Thus, row 515 includes frames f-1 to f-14, row 516 includes frames f-15 to f-28, row 517 includes frames f-28 to f-42, row 518 includes frames f-43 to f-56, row 519 includes frames f-57

to f-70, row 520 includes frames f-71 to f-84, row 521 includes frames f-85 to f-98, row 522 includes frames f-99 to f-112, and row 523 includes frames f-113 to f-126.

[0092] **Figure 5B** illustrates exemplary compression paths for the block B-1. For example, compress frame f-16, the delta frame between f-16 and f-1 to get frame f-1; compress frame f-16 and the delta frame between f-16 and f-2 to get frame f-2; compress frame f-16 and the delta frame between f-16 and f-3 to get frame f-3; and, compress frame f-16, the delta frame between f-16 and f-17, and the delta frame between f-17 and f-4 to get frame f-4. Next, compress frame f-16 and the delta frame between f-16 and f-15 to get frame 15; compress frame f-16 to get frame f-16; compress frame f-16 and the delta frame between f-16 and f-17 to get frame f-17; compress frame f-16, the delta frame between f-16 and f-17, and the delta frame between f-17 and f-18 to get frame f-18. Next, compress frame f-16 and the delta frame between f-16 and f-29 to get frame f-29; compress frame f-16 and the delta frame between f-16 and f-30 to get frame f-30; compress frame f-16 and the delta frame between f-16 and f-31 to get frame f-31; compress frame f-16, the delta frame between f-16 and f-17, and the delta frame between f-17 and f-32 to get frame f-32. Lastly, compress frame f-16, the delta frame between f-16 and f-30, and the delta frame between f-30 and f-43 to get frame f-43; compress frame f-16, the delta frame between f-16 and f-30, and the delta frame between f-30 and f-44 to get frame f-44; compress frame f-16, the delta frame between f-16 and f-30, and the delta frame between f-30 and f-45 to get frame f-45; and, compress frame f-16, the delta frame between f-16 and f-31, and the delta frame between f-31 and f-46 to get frame f-46.

[0093] The compression paths for the remaining block B-2, B-3, B-4, B-5, and B-6 are similar to B-1 (see **Figure 5B**) discussed above.

[0094] As previously mentioned, storing all of the compression sequences together with individual compressed data for each frame for the blocks B1-B6 having 126 frames will result in a large final file. To minimize the amount of data to be stored and the amount of repeated compressions for the redundant frames the compressor of this embodiment compresses the key frame and the redundant delta frames only once. For instance, the frames f-16, f-21, and f-26 would not be compressed multiple times just because they appear several times in several compression paths, but instead, each of these frames would be compressed only once. The first time each of these frames is compressed, the compressor would have memorized that these frames have already been compressed once. The next time any of those frames that were already compressed once is needed again, the compressor will skip the compression of that frame, and use the results of the previous compression.

[0095] In one embodiment, a video track 550 and a data source track 570 as shown in **Figure 5C** are generated. This video track 550 describes the compression (or decompression) sequences for each of the frame in each of the blocks. This will enable quick random access to each of the frame in movie object. For instance, when a particular frame is requested, the sequences will indicate what non-key frames and what key frame are needed before the particular frame is available. The data source track 570 stores the compressed key frame and the compressed non-key frames. Each of the compressed frames can be referred to multiple times.

[0096] A reference table (not shown) or a set of reference links 551 is generated to allow each of the frame in the blocks B-1 to B-6 of the object movie track 550 to reference the appropriate compressed data in the data source track 570. In this sense, the

video track 550 only contains the order or the sequence to compress and decompress each of the frame but not the actual compressed data themselves. The data source track 570 comprises all of the actual compressed data. As discussed above, each of the frames in the object movie is only compressed once and is stored in the data source track 570. The links 551 can be used to refer to the compressed data in the data source track 570 as many times as necessary. To get the compressed data for any one frame, the links 551, for example, would link the frames in the video track 550 to the appropriate actual compressed data which can then allow the frame to be accessed for decompression.

[0097] Another way to store the compressed data is to put all the key frames at the beginning of the data track 570-1. (See **Figure 5C-1**). As illustrated in **Figure 5C-1**, a section S-1 contains the compressed data for all of the key frames, (f-16, f-21, f-26, f-91, and f-96), in the object movie described in **Figure 5B**. A section S-2 contains the compressed data for all of the non-key frames in the block B-1; a section S-3 contains the compressed data for all of the non-key frames in the block B-2; a section S-4 contains the compressed data for all of the non-key frames in the block B-3; a section S-5 contains the compressed data for all of the non-key frames in the block B-4; a section S-6 contains the compressed data for all of the non-key frames in the block B-5; and a section S-7 contains the compressed data for all of the non-key frames in the block B-6. In this example, the user can get an outline view of the object while waiting for the rest of the non-key frame data, coming from, for example, the Internet.

[0098] As an illustration, when the compressor compresses frame f-16, the delta frame between f-16 and f-17, and the delta frame between f-17 and f-4 to compress f-4, the compressor will memorize that frame f-16, f-17 and f-4 have been compressed.

When the compressor continues to compress subsequent frames, the compressor will skip the compression that requires the compression of frames f-16, f-17 and f-4 again. To compress f-18, the compressor will skip compression for frames f-16, f-17, and only compress the delta frame between f-17 and f-18. The similar processes continue until all of the frames are compressed. Then, to randomly access any one frame, (e.g., f-18) the decompressor would ascertain the sequence of decompression of that frame by examining the video track 550 (e.g., f-16, f-17, and f-18). Using the links 551, the video track 550 will lead the decompressor to the location in the data source track 570 that contains the compressed frames and delta frames needed to decompress the particular frame, frame f-18.

[0099] Similar to the video track 550 and the data source track 570 described above, additional location and compressed data is added to the video and a data source track for blocks B-2, B-3, B-4, B-5, and B-6 to complete the compression of the object movie. Note that there is only one track created for all blocks.

[0100] **Figure 6** illustrates a compression method 600 of an object movie. The movie object comprises a series of images or views of any one object in some sequential order. Examples of what constitutes an object in an object movie includes anything such as a person, a thing, a scene, a setting, a setting with many things included within the setting, or a group of objects.

[0101] In step 602, after a set of views of the object has been captured, for example, by using the method described in **Figure 3A**, the object movie is divided into blocks of specified size. In one example, the object movie is divided such that each of the blocks represents a particular region of the object. In another example, each of the

blocks is proximately associated with each other. In yet another example, the block size and location are selected such that the frames within each of the blocks share spatial similarities to adjacent frames.

[0102] In step 604, each of the blocks is compressed separately. One method of compression is to determine at least one key frame for each block and compress the frames in each blocks based on the key frame selected. In one example, frame differencing is used for the compression technique. In another example, frame prediction is used for the compression technique. These techniques, frame differencing compressions, are well known in the art.

[0103] In step 606, a video track layout is generated for the compression sequences for each of the frames and each of the blocks of the object movie. The video track layout would make references to a data source track that stores all of the compressed frames. In one example, each of the frames is only compressed once. The video track layout would then refer to the data source track multiple times and as many times as necessary to access the compressed frames for any particular sequence in the video track.

[0104] **Figure 7** shows an example of a method 700 which, may be performed by a transmitting system according to the present invention. In one embodiment, the transmitting system is a server computer system like the system 109 and the system 151 described above. The method 700 can be a digital processing implemented method for transmitting a data set that enables interactive random access to different views of an object. The method 700 begins in step 702 in which, various different views are generated or captured for an object at various perspectives. This may involve the

computer generation of various images or views of the object or may involve the capturing through a digital camera or an analog camera of various views of the object. In one example, the various images or views may be created as discussed in **Figure 3A**.

[0105] In step 704, the frames are arranged in a preferred order and layout. For example, if the object is a car, the bottom views of the car may be placed in the beginning of the preferred order in one set of the views having a first preferred order which is designed for car mechanics, and top or side views may be placed in the beginning of another preferred order which is designed for normal consumers and possible purchasers of the car. The preferred layout may be a two-dimensional array arranging frames in a table like format having rows and columns as shown in **Figure 5A**. The rows of the frames can be the horizontal views and the columns of the frames can be the vertical views of the object. The preferred layout may also take on other types of array shape, for example, a triangle.

[0106] As discussed above, one way to create the horizontal views is by panning the camera around the object with some degrees of increment, for example, a 10-degree increment. In the example shown in **Figure 5A**, the row 515 could represent the top horizontal views of the object. The row 519 could represent the horizontal views of the mid-plane of the object. The row 523 could represent the horizontal views of the bottom plane of the object. One way to create the vertical views is by tilting the camera around the object from the top of the object to the bottom with some degrees of increment, for example, a 10 degrees increment. The column 501 could represent the vertical views of the object to provide preference to certain views, which may be referred to as key views, when transmitting the views. For example, the views are arranged for transmission in a

preferred order which is based upon which views would be considered to be the most important to see first by a viewer of the object.

[0107] In step 706, the preferred layout is divided into a plurality of blocks. In one example, each of the blocks is equal to or smaller than a predetermined or specified size. The preferred layout should be divided so that each of the blocks represents particular regions of the object. And, the size of each of the blocks should be such that the frames in the block share spatial similarities to each other.

[0108] In step 708, at least one key frame is determined for each of the blocks. Preferably, the key frame is the center frame of the block. If there is no center key frame, the frame closest to the center of the block should be the key frame. Furthermore, the key frame should be closer to the top left corner of the block.

[0109] In step 710, the compression paths for each of the blocks are determined. (1) Always start compressing with the key frame. (2) Always sweep away from the key frame. (3) Sweep horizontally or vertically from the key frame until reaching the boundaries of the object movie to cover all of the frames possible in the horizontal path and the vertical path. (4) From the horizontal path and the vertical path of each frame, sweep diagonally until reaching the boundaries to cover all of the frames not reached by the horizontal path and vertical path. And, (5) always sweep such that the compression paths reach every one of the frames in the object movie. **(See illustration in Figure 5B).**

[0110] In step 712, a link table is generated. In step 714, the key frame is compressed first and then the delta frames for the non-key frame in each of the block are compressed. Frame differencing compression method can be one compression method used to perform the compression. Compress each key frame and each delta frames for

the particular non-key frame only once. During compression, if the key frame has already been compressed once and is then encountered again in subsequent compression paths, (e.g., when compressing along the sequence in the video track), skip compression of that key frame. Similarly, during compression, if the delta frames for the particular non-key frames have already been compressed once and are encountered again in subsequent compression paths, skip compression of those delta frames.

[0111] Then, in step 716, a data source track is generated which comprises the compressed key frame and the compressed delta frames for the object movie. The step 716 generates a data source track that stores all of the compressed key frames and non-key frames for all of the blocks. The video track will refer to the compressed data in the data source track to access any compressed frame and delta frames for the non-key frames. Each of the frame and delta frames can be referred to as often as they are needed in the video track. A table of reference or a set of links is generated so that the compression sequences in the video track can refer to the compressed frames in the data source track. The links can be stored together with the video track generated in step 712.

[0112] After the data source track is generated, a video track (the reference track) is generated for all of the blocks (see step 718). A video of the object movie is then generated based on the link table and the data track.

[0113] **Figures 8A-8B** illustrate a method 800 performed by a client system according to the present invention. It will be appreciated that various steps are shown and the particular sequences of **Figures 8A-8B** are only for purposes of illustration. Certain of these steps may be performed in a different order, such as step 814, step 812, and step 816. The method 800 is a digital processing implemented method for receiving

and using a data set that enables interactive random access for different views of an object in an object movie. Interactive random access is typically requested by a user of the client system in which the user randomly chose to view any particular views of the object in any particular order.

[0114] In step 802, a request is transmitted to a host system, for example, a web server 109. A client system such as client system 121, 125, 135, or 137 can be used to transmit such a request. The client system transmits a request to receive the data set representing an object that has multiple views for viewing. Typically, this transmission will be considered to be a download from a server system. The client system, in one example of the present invention, may specify a particular set of preferred orders if more than two sets of preferred orders are available for downloading. So, for example, if a first set of preferred order is designed to provide preference to the top and bottom views, another set of preferred orders may provide preference to the side views of the object.

[0115] Then, in step 804, the client system receives the object movie wherein the data, in a preferred order and layout, is compressed in separate blocks. For instance, the object movie can be divided into blocks having two-dimensional arrays as discussed above. Then, each of the blocks is compressed separately according to some of the embodiments discussed above. The sequence of the compressed data can be of different order. In one example, all key frame data can be saved at the start of a data track followed by all the non-key frame data, which can be grouped in blocks (as illustrated in **Figure 5C-1**). In another example, the key frame and non-key frame data are grouped in a block and saved in the data track as such.

[0116] Step 806 allows the user to manipulate the transmitted object movie. For example, step 806 allows the user to move a cursor relative to a display of the compressed object movie. The manipulation may occur by selecting a different view by rotating the object with the cursor. Example of a method of manipulation the object can be found in U.S. Patent No. 5,019,809 by Michael Chen. This patent describes a method for direct manipulation of an object by using a two-dimensional cursor control device such as a mouse, to simulate three-dimensional movement of the object. This enables the user to see different views of the object as well as to rotate or otherwise manipulate the object in order to see these different views.

[0117] Step 808 maps the cursor to the appropriate frame number in a video track that comprises the sequence of compression and decompression for each of the frame of the object movie generated according to some of the embodiments discussed above. Step 808 maps the cursor to the appropriate frame upon receiving some input that requests a selected umber of views of the object movie. The user typically issues the input during the user's interacting with the object movie as discussed in step 806. The video track also includes data as to how to locate the compressed frames for each of the views or images of the object movie. If data of a frame are not available, e.g., not yet downloaded from the Internet, another available frame that is the closet to that not-yet-available frame can be chosen.

[0118] Step 810 uses the frame number and the corresponding sequence information in the video track to find the appropriate links which in turn lead to the actual compressed data of the selected images or views. As mentioned above, the user may use

the cursor to select particular views of the object that the user wishes to see. These views are referred to as selected images or selected views.

[0119] Step 812 will then decompress the appropriate compressed data and display the selected images or views, for example, by displaying them on a monitor display device 161. Furthermore, step 814 (**Figure 8B**) stores all of the selected frames that have been decompressed.

[0120] The aforementioned methods enable quick and convenient user interactive activities. The decompression time for any frame is short thus giving a quick response time. Also, the compressed file remains small because these methods can use a small number of key frames even with the multi directional compression paths. That helps to give the object movie a fast transmission rate during export and access.

[0121] It should also be noted that the present invention may be used with the transmission of object data from one system to another system where there is no network connection but where the transfer of data is at a slow rate. Furthermore, the present invention may be used in transferring data from a slow storage device which provides a slow data transmission rate or from a device which does not provide random access.

[0122] **Figures 9-10** illustrate an example according to one embodiment of the present invention for two different computer readable storage media. It will be appreciated that the actual memory which stores this information may be different elements, such as memory 155 and mass memory 163, or they may be the same element, such as mass memory 163. In one example of a network computer where there is no non-volatile mass storage, the necessary software files and data files may be downloaded to the memory 155 for execution in a processor of the network computer. In this case, the

memory 155 provides the computer readable storage medium. In one embodiment, the client computer system of the server computer system's computer readable media may at some time be entirely stored in a nonvolatile mass memory, such as a hard disk. At other times, the various elements shown in either **Figure 9** or **Figure 10** may be dispersed between dynamic memory, such as a memory 155, and mass memory, such as memory 163.

[0123] **Figure 9** shows a client system's computer readable storage medium 900. the client system's computer readable storage medium stores executable computer program instructions which cause a client computer system to perform a method of the present invention. This medium includes software routines 902 for providing an operating system for the client, client OS. View display and view selector routines 904 provide the necessary software to display the various views and to select a view or views that the user selects. Network transmission routines 906 provide the various software routines necessary to request, transmit, and receive the data, such as TCP/IP protocols. The view manipulation routines 908 include the necessary software routines to allow the user to manipulate the object in order to see various different views of the object. The view tracking routines and the view decompression routines 910 include the software routines which track the data source track for the views selected by the user so that these views can be decompressed and transmitted to the user. The medium 900 may also include software routines which allow the playback of a sequence or views, (e.g., the sequence chosen by the user), as a movie.

[0124] **Figure 10** illustrates a server system's computer readable storage medium for storing executable program instructions which cause the server computer system to

perform a method of the present invention. The computer storage readable medium 1000 includes a server operating system component 1002 which provides an operating system for the server computer, server OS. The views of the object 1004 are also stored or are accessible to the server computer system. Typically, the view of the object 1004 are stored on some storage device which may be retrieved by the server computer system, and the object data is prearranged in the preferred order or orders and layout. As noted above, this prearrangement may exist by a table which specifies the preferred order rather than by actually storing the data in a preferred order. Network transmission routines 1006 provide the software which is used to transmit data from the server and to receive instructions and other data from a client system. Data component 1008 includes information regarding the views about the particular object. This information may be provided to a client system in order to allow the client system to display the process of a transmission through an indicator, such as the indicator appearing on the display.